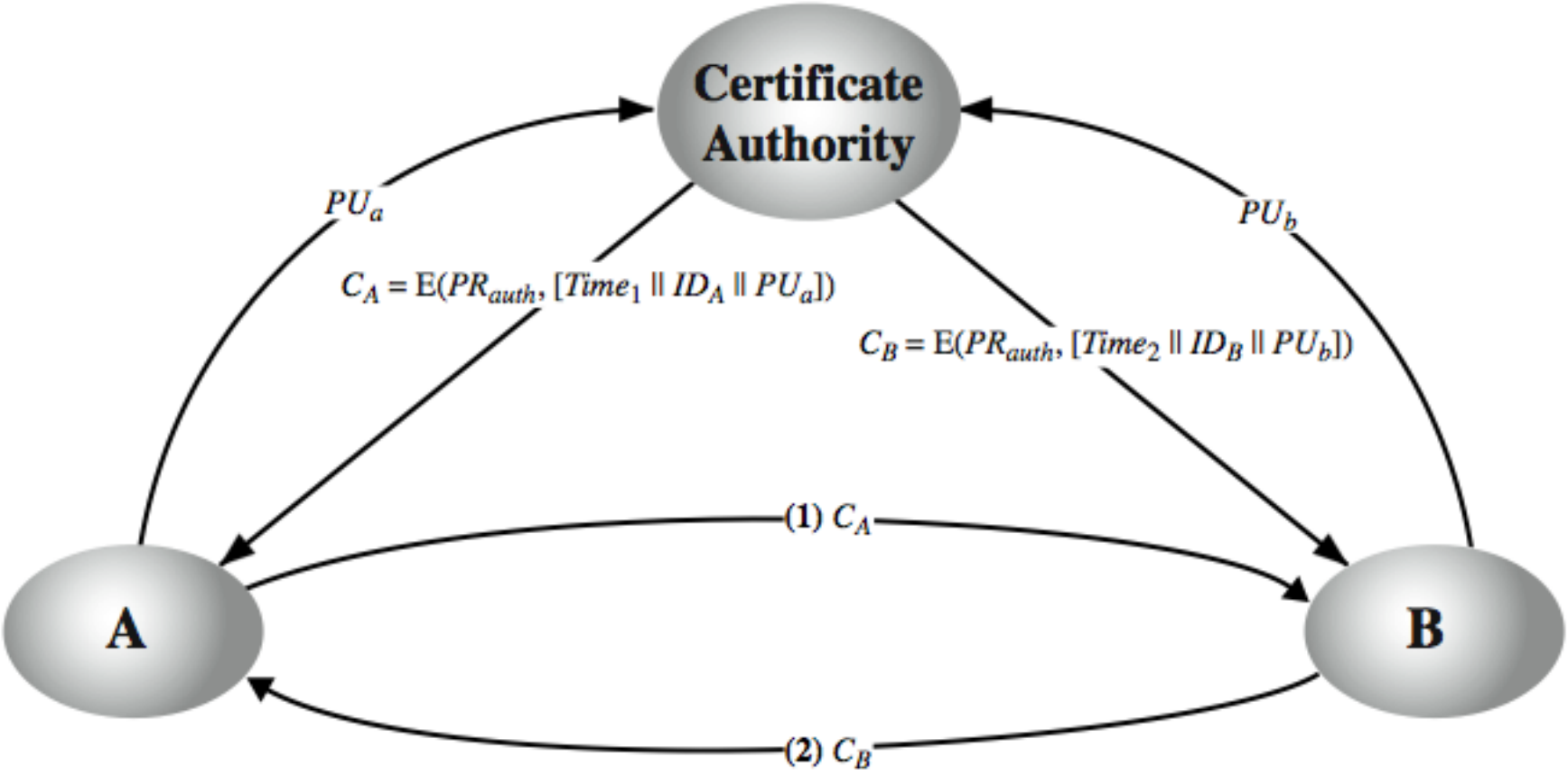
Chapter 14

Key Management
Public-Key Certificates Scheme

Public-Key Certificates

- certificates allow key exchange without real-time access to public-key authority
- a certificate binds **identity** to **public key**
  - usually with other info such as period of validity, rights of use etc
- with all contents **signed** by a trusted Public-Key or Certificate Authority (CA)
- can be verified by anyone who knows the public-key authorities public-key

Public-Key Certificates



1. A sends a message to the public-key authority containing a request for the current public key of B.

2. The authority responds with a message that is encrypted using the authority's private key, PR_{auth} Thus, A is able to decrypt the message using the authority's public key. Therefore, A is assured that the message originated with the authority. The message includes the following:
   - B's public key, PU_b which A can use to encrypt messages destined for B
   - The original request, to enable A to match this response with the corresponding earlier request and to verify that the original request was not altered before reception by the authority

3. A stores B's public key and also uses it to encrypt a message to B

4. B retrieves A's public key from the authority in the same manner as A retrieved B's public key. At this point, public keys have been securely delivered to A and B, and they may begin their protected exchange.

Public-Key Certificates

- Each participant applies to the certificate authority, supplying a public key and requesting a certificate.
- For participant A, the authority provides a certificate CA. A may then pass this certificate on to any other participant, who can read and verify the certificate by verifying the signature from the certificate authority.
- Because the certificate is readable only using the authority's public key, this verifies that the certificate came from the certificate authority.
- A generates a new private/public key pair and applies to the certificate authority for a new certificate.

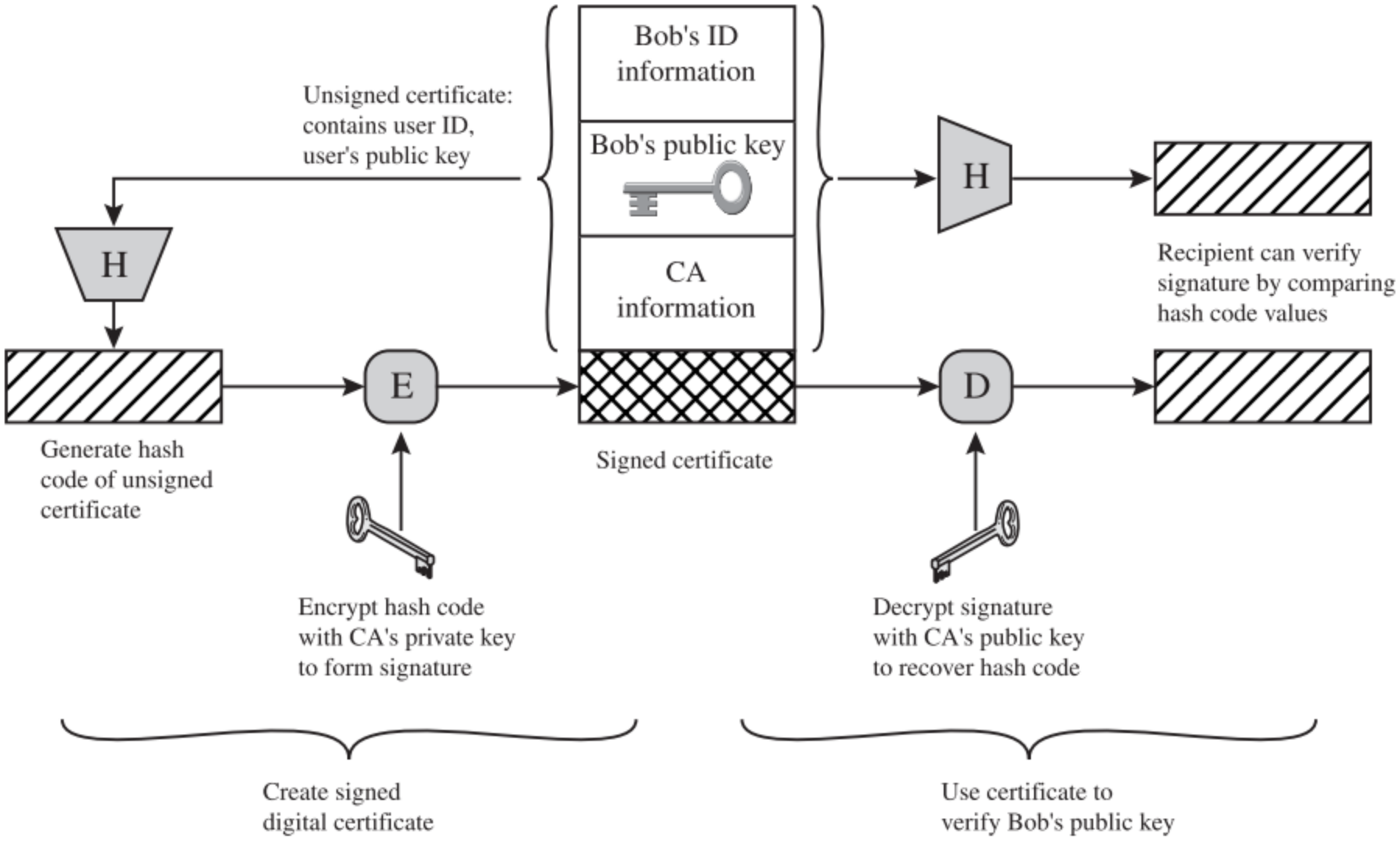
Requirements on Public-Key Certificates Scheme

- Any participant can read a certificate to determine the name and public key of the certificate's owner.
- Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
- Only the certificate authority can create and update certificates.
- Any participant can verify the currency of the certificate.

X.509 Authentication Service

- part of the X.500 series of recommendations that define a directory service, being a server or distributed set of servers that maintains a database of information about users.
- defines framework for authentication services
  - directory may store public-key certificates
  - with public key of user signed by certification authority
- also defines authentication protocols
- uses public-key crypto & digital signatures
  - algorithms not standardised, but RSA recommended
- X.509 certificates are widely used
  - have 3 versions

X.509 Certificate Use



X.509 Certificates

Version: Differentiates among successive versions (1,2,3) of the certificate format.

Serial number: An integer value unique within the issuing CA.

Signature algorithm identifier: The algorithm used to sign the certificate together with any associated parameters.

Issuer name: the name of the CA that created and signed this certificate.

Period of validity: Consists of two dates: the first and last on which the certificate is valid.

Subject name: The name of the user to whom this certificate refers .

Subject's public-key information: The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.

Issuer unique identifier: (optional) used to identify uniquely the issuing CA.

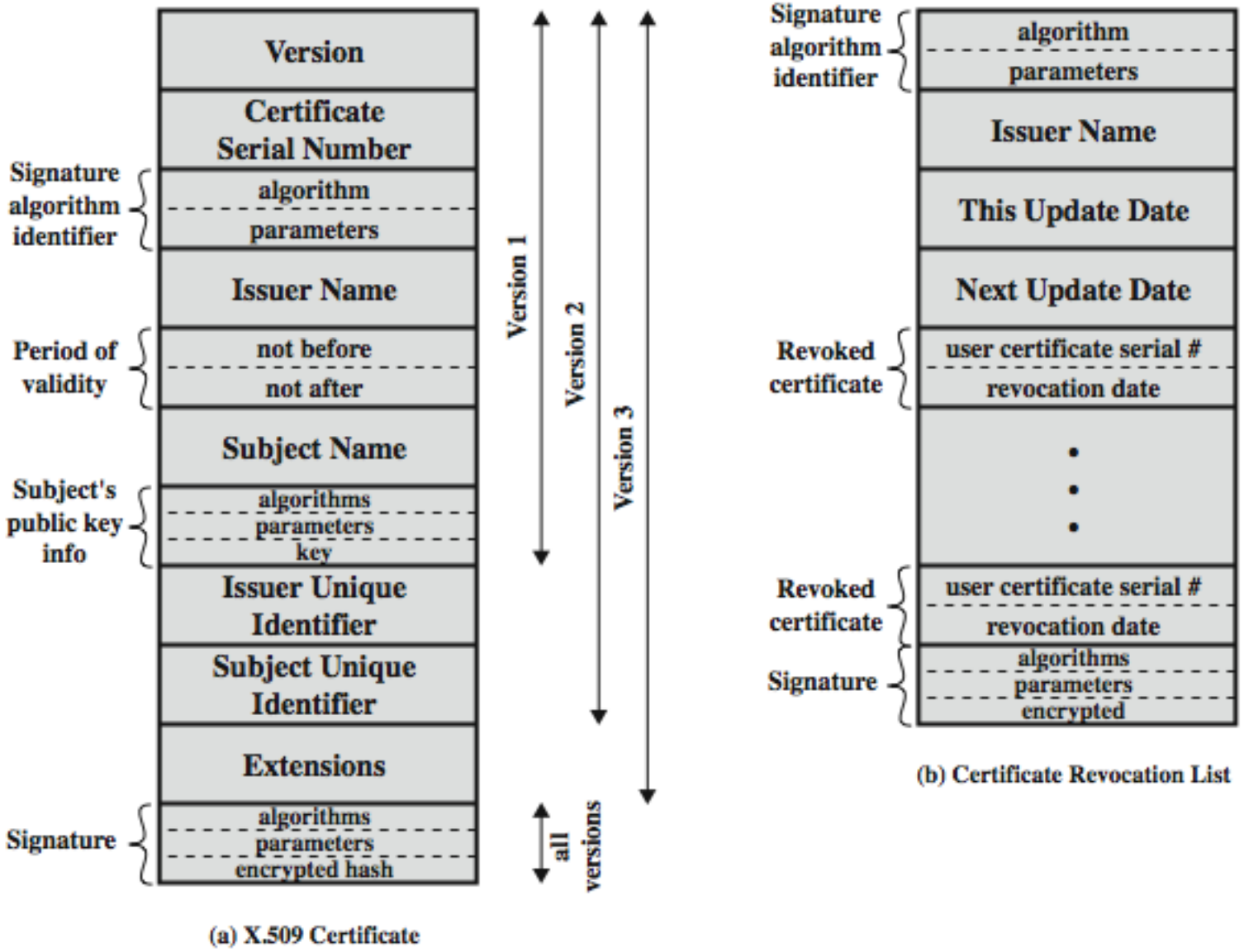
Subject unique identifier:(optional)used to identify uniquely the subject.

Extensions: A set of one or more extension fields.

Signature: it contains the hash code of the other fields encrypted with the CA's private key . This field includes the signature algorithm identifier.

Y<<X>> = the certificate of user X issued by certification authority Y

X.509 Certificates



(a) X.509 Certificate

(b) Certificate Revocation List

Obtaining a Certificate

- any user with access to CA can get any certificate from it
- only the CA can modify a certificate
- because cannot be forged, certificates can be placed in a public directory

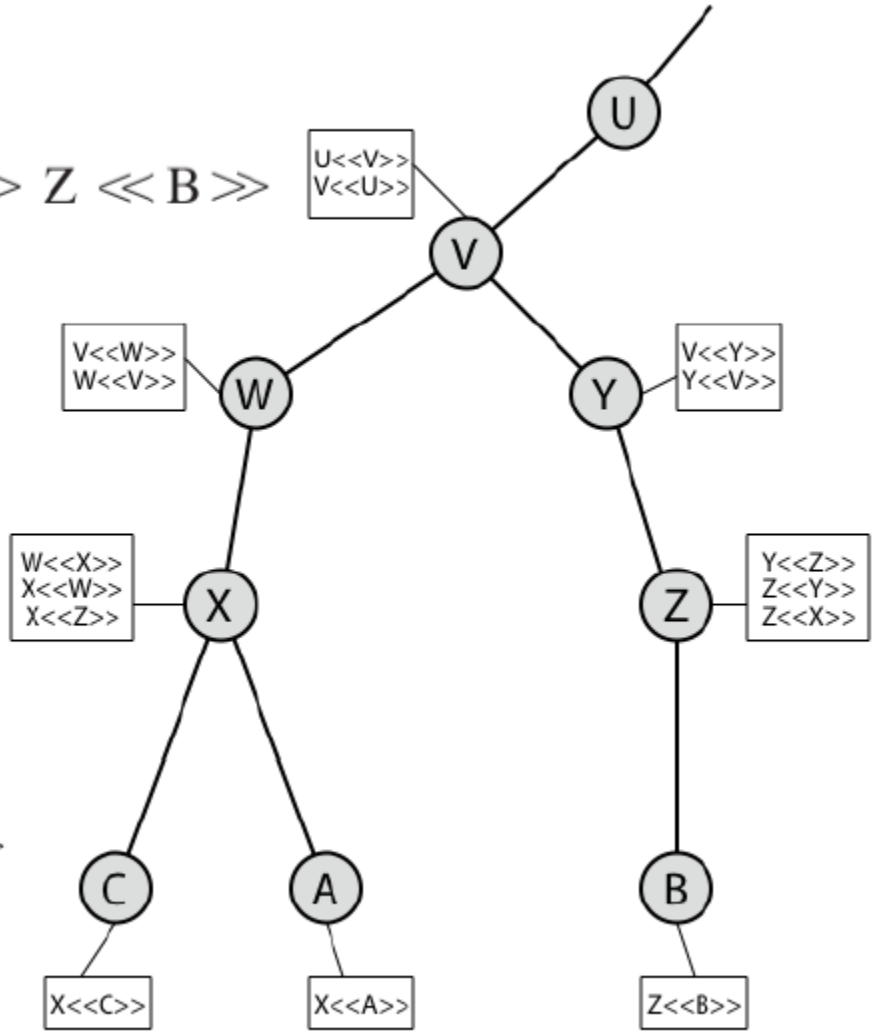
CA Hierarchy

- if both users share a common CA then they are assumed to know its public key
- otherwise CA's must form a hierarchy
- use certificates linking members of hierarchy to validate other CA's
  - each CA has certificates for clients (forward) and parent (backward)
- each client trusts parents certificates
- enable verification of any certificate from one CA by users of all other CAs in hierarchy

CA Hierarchy Use

A can acquire the following certificates from the directory to establish a certification path to B:

$$X \ll W \gg \; W \ll V \gg \; V \ll Y \gg \; Y \ll Z \gg \; Z \ll B \gg$$



B obtains A's public key from the following certification path:

$$Z \ll Y \gg \; Y \ll V \gg \; V \ll W \gg \; W \ll X \gg \; X \ll A \gg$$

The directory entry for each CA includes two types of certificates:
Forward certificates: Certificates of X generated by other CAs.
Reverse certificates: Certificates generated by X that are the certificates of other CAs.

Certificate Revocation

- certificates have a period of validity
- may need to revoke before expiry, eg:
  1. user's private key is compromised
  2. user is no longer certified by this CA
  3. CA's certificate is compromised
- CA's maintain list of revoked certificates
  - the Certificate Revocation List (CRL)
- users should check certificates with CA's CRL